

# Spesifikasi Sistem Multi Agen Berparameter Secara Formal dengan TLA+

Dr.rer.nat. Cecilia E. Nugraheni,S.T., M.T.

# Daftar Isi

<b>1</b>	<b>Pendahuluan</b>	<b>1</b>
1.1	Latar Belakang . . . . .	1
1.2	Rumusan Masalah . . . . .	3
1.3	Tujuan . . . . .	3
1.4	Desain dan metode penelitian . . . . .	4
1.5	Batasan Penelitian . . . . .	4
1.6	Sistematika Pembahasan . . . . .	5
<b>2</b>	<b>Spesifikasi Sistem Berparameter</b>	<b>6</b>
<b>3</b>	<b>Studi Kasus 1: Reader-writer protocol</b>	<b>8</b>
<b>4</b>	<b>Studi Kasus 2: Block-world problem</b>	<b>12</b>
<b>5</b>	<b>Penutup</b>	<b>16</b>

# Daftar Gambar

3.1	Spesifikasi untuk <i>reader writer protocol</i> . . . . .	10
4.1	Contoh dari <i>block-world problem</i> . . . . .	13
4.2	Spesifikasi untuk block-world problem. . . . .	15

## Abstrak

Sebuah Sistem Multi Agen (SMA) adalah kumpulan agen cerdas yang berinteraksi satu dengan yang lain dan bekerja secara bersama-sama untuk mencapai suatu tujuan. Pada penelitian SMA dipandang sebagai suatu sistem berparameter yaitu sistem yang terdiri atas beberapa proses sejenis yang banyaknya ditentukan oleh parameter masukan. Motivasi dari penelitian ini adalah dengan memandang SMA sebagai suatu sistem berparameter, maka proses spesifikasi dan verifikasi dapat dilakukan dengan cara yang sama berapapun jumlah agen yang ada di dalam sistem.

Permasalahan yang mungkin muncul dengan cara pandang ini adalah tidak semua SMA bersifat homogen, tidak semua agen mempunyai kapabilitas yang sama. Pada penelitian ini diusulkan sebuah solusi sederhana yaitu dengan menambahkan prekondisi pada aksi pada spesifikasi SMA yang berbasis Temporal Logic of Actions (TLA). Hal ini untuk menjamin hanya agen tertentu yang dapat mengaktifkan aksi tertentu. Pendekatan ini telah berhasil diterapkan pada SMA homogen maupun non-homogen.

*Kata kunci: multi agent systems, parameterized systems, specification, TLA.*

## Abstract

A multi agent system is a collection of intelligent agents that interact with each other and work together to achieve a goal. In this paper we view multi agent systems as parameterized systems which are systems that consist of several similar processes whose number is determined by an input parameter. The motivation of this work is that by treating multi agent systems as parameterized systems, the specification and verification processes can be done in the same way regardless of the number of agents involved in the multi agent systems.

Problem that may arise due to this treatment is that not all multi agent systems are homogeneous, i.e. not every agent has the same capability. This paper presents a simple solution for this problem which is by adding preconditions to actions in the TLA-based specification to ensure that only the appropriate agent can take a particular action. This approach is successfully applied to a homogeneous and a non-homogeneous multi agent system.

*Kata kunci: multi agent systems, parameterized systems, specification, TLA.*

# Bab 1

## Pendahuluan

### 1.1 Latar Belakang

Sistem Multi Agen (SMA) dimengerti sebagai kumpulan agen cerdas, dalam hal ini adalah perangkat lunak atau program, yang saling berinteraksi dan bekerja sama untuk mencapai suatu tujuan. Karakteristik umum dari SMA adalah [13]:

1. setiap agen mempunyai informasi dan kemampuan menyelesaikan masalah yang tidak lengkap
2. tidak ada kendali sistem yang sifatnya global
3. data terdesentralisir pada masing-masing agen
4. komputasi bersifat asinkron

Penelitian yang mengkaji SMA sudah banyak dilakukan, tidak terkecuali tinjauan dari sudut metode formal. Beberapa publikasi yang membahas tentang pembuatan spesifikasi dan verifikasi secara formal adalah:

1. Pada [8] penulis mengetengahkan permasalahan pembatasan hak akses dari agen-agen pada suatu SMA. Pembatasan hak akses ini akan menjadi kompleks pada SMA yang besar. Untuk itu diperlukan cara perhitungan. Pendekatan yang diusulkan adalah menggunakan metode tablo.
2. Pada [13] disajikan sebuah pendekatan formal untuk memverifikasi analisis tugas kognitif, yaitu sebuah kakas analitik yang telah berhasil digunakan dalam merancang perilaku reaktif pada arsitektur multi-agen.
3. Szirbik dan Wagner menerapkan konsep SMA pada aplikasi *e-business* [15]. Pada makalah mereka dibahas tentang spesifikasi, khususnya tipe-tipe interaksi proses untuk aplikasi *e-business* dan diperkenalkan konsep koherensi untuk interaksi proses dalam konteks komunikasi multi-agen yang lebih luas. Spesifikasi dibuat dengan menggunakan aturan reaksi dan analisisnya dilakukan dengan menggunakan Petri net.

Pada penelitian ini, SMA dipandang sebagai suatu sistem berparameter yaitu sistem yang terdiri atas sejumlah proses sejenis yang banyaknya ditentukan oleh sebuah parameter masukan. Diharapkan bahwa dengan memperlakukan SMA sebagai suatu sistem berparameter, proses spesifikasi dan verifikasi dapat dilakukan dengan cara yang sama tanpa dipengaruhi banyaknya proses yang terlibat pada SMA.

Terdapat berbagai cara untuk mendeskripsikan atau menspesifikasikan SMA ini. Pada umumnya, setiap agen akan direpresentasikan sebagai sebuah modul program. Karena setiap agen mungkin mempunyai atribut yang berbeda satu dengan yang lain, begitu pula dengan fungsinya, maka modul antara satu agen dengan agen yang lain bisa berbeda. Banyaknya modul yang harus dibuat akan berbanding lurus dengan banyaknya agen. Semakin besar SMA maka akan semakin banyak pula modul yang harus disediakan. Dengan bervariasinya tugas dan kemampuan agen, maka

kompleksitas SMA akan tinggi.

Pada penelitian ini ditinjau sebuah kelas dari SMA yang diberi nama Sistem Multi Agen Berparameter (SMAB). Yang dimaksud dengan SMAB di sini adalah sebuah SMA dimana agen-agennya seragam atau mirip satu dengan yang lain. Parameter dalam konteksnya ini adalah sebuah bilangan bulat positif yang menyatakan banyaknya agen.

Penelitian ini difokuskan pada pembuatan spesifikasi dari SMAB secara formal. Bahasa spesifikasi yang akan digunakan adalah TLA+.

Sejauh kajian pustaka yang telah dilakukan, belum ada yang mengkaji SMA sebagai suatu sistem berparameter. Untuk itu diharapkan bahwa penelitian yang diusulkan ini dapat memberikan kontribusi bagi bidang kajian SMA secara umum.

## **1.2 Rumusan Masalah**

Masalah yang diangkat pada penelitian ini dapat dirumuskan sebagai berikut:

1. SMA seperti apa yang dapat dikategorikan kedalam SMAB?
2. bagaimana membuat spesifikasi SMAB dalam notasi TLA+?

## **1.3 Tujuan**

Berdasarkan permasalahan yang muncul, maka penelitian ini bertujuan untuk melihat sejauh mana SMA dapat diperlakukan sebagai SMAB dan melihat sejauh mana SMA dapat dinyatakan dengan menggunakan format spesifikasi TLA+ standar untuk sistem berparameter.



## 1.4 Desain dan metode penelitian

Langkah-langkah yang dilakukan pada penelitian ini adalah sbb.:

1. Studi literatur
2. Pencarian beberapa studi kasus SMA
3. Melakukan analisis terhadap studi kasus, sejauh mana dapat diperlakukan sebagai sistem berparameter.
4. Membuat kerangka spesifikasi umum untuk SMAB dalam notasi TLA+
5. Menggunakan kerangka spesifikasi yang dibuat untuk studi kasus

Pada penelitian ini akan ditinjau dua buah studi kasus, yaitu *reader-writer protocol* dari [10] dan *block-world problem* dari [14]. *Reader-writer protocol* mewakili SMA homogen, sementara *block-world problem* mewakili SMA non-homogen. Untuk menuliskan spesifikasi akan digunakan Temporal Logic of Action (TLA+) dari Lamport [7].

## 1.5 Batasan Penelitian

Penelitian ini lebih ditekankan pada sejauh mana sebuah SMA dapat dinyatakan sebagai suatu sistem berparameter. Untuk penyederhanaan permasalahan tidak akan ditinjau beberapa sifat yang harusnya menjadi properti dari SMA seperti yang disebutkan pada bagian Latar Belakang.

## 1.6 Sistematika Pembahasan

Laporan penelitian ini terdiri atas lima bab. Bab pertama adalah Pendahuluan yang berisi tentang masalah penelitian, tujuan, desain dan metode penelitian, dan sistematika pembahasan. Bab 2 menjelaskan dengan singkat spesifikasi umum untuk sistem berparameter. Bab 3 membahas kasus studi untuk SMA homogen, yaitu *reader-writer protocol*. Bab 4 membahas kasus studi untuk SMA non-homogen yaitu *block-world problem*. Bab 5 adalah bab penutup yang berisi kesimpulan dan saran.

## Bab 2

# Spesifikasi Sistem Berparameter

Pada penelitian ini, diambil batasan sistem berparameter yang ditinjau adalah sistem berparameter yang sifatnya *interleaving*, yaitu pada setiap saat hanya ada satu subsistem yang aktif, dan terdiri atas sejumlah terbatas komponen diskret. Misal  $M$  menyatakan himpunan tak kosong yang berisi proses yang sedang berjalan pada sistem yang ditinjau. Sebuah sistem berparameter dapat dituliskan sebagai formula berbentuk:

$$parSpec \equiv \forall k \in M : Init(k) \wedge \Box[Next(k)]_{v[k]} \wedge L(k). \quad (2.1)$$

dimana

- $Init$  adalah sebuah predikat status yang menyatakan kondisi awal sistem secara global,
- $Next(k)$  adalah sebuah aksi yang mencirikan relasi *next-state* dari suatu proses  $k$ ,

- $v$  adalah sebuah fungsi status yang menyatakan variabel sistem dan
- $L(k)$  adalah sebuah formula yang menyatakan kondisi *liveness* yang diharapkan dari proses atau subsistem  $k$ .

## Bab 3

# Studi Kasus 1: Reader-writer protocol

Studi kasus yang pertama adalah *reader-writer protocol* [10]. Protokol ini dapat dijelaskan secara ringkas sebagai berikut: Terdapat  $M$  proses yang berjalan pada sistem dan setiap proses dapat melaksanakan aksi berikut:

- $ImmRd(k)$ : Proses  $k$  memberi sinyal kalau akan membaca. Jika tidak ada proses yang sedang atau akan menulis, maka proses  $k$  langsung diijinkan untuk membaca data yang diinginkan.
- $ReqRd(k)$ : Proses  $k$  meminta akses untuk membaca, tetap terdapat proses lain yang sedang atau akan menulis, Maka  $k$  dimasukkan kedalam daftar tunggu proses yang akan melakukan pembacaan.
- $GrRd(k)$ : Jika tidak proses yang melakukan penulisan, maka proses  $k$ , yang berada dalam daftar tunggu proses yang akan melakukan pembacaan, diijinkan untuk melakukan pembacaan.

- $EndRd(k)$ : Proses  $k$  yang sedang melakukan pembacaan menyatakan telah selesai membaca dan meninggalkan protokol.
- $ImmWr(k)$ : Proses  $k$  memberi sinyal menyatakan keinginan untuk menulis. Jika tidak ada proses lain yang sedang membaca atau menulis maupun tidak ada proses yang sedang menunggu untuk menulis, proses  $k$  diijinkan untuk melakukan penulisan.
- $ReqWr(k)$ : Proses  $k$  memberi sinyal bahwa akan melakukan penulisan, tetapi prekondisi untuk aksi  $ImmWr(k)$  tidak terpenuhi. Maka proses  $k$  dimasukkan ke dalam daftar tunggu proses yang akan melakukan penulisan.
- $GrWr(k)$ : Proses  $k$  yang menunggu giliran untuk menulis diijinkan untuk melakukan penulisan jika tidak ada proses lain yang sedang atau akan membaca dan tidak ada proses yang sedang menulis.
- $EndWr(k)$ : Proses  $k$  yang telah melakukan penulisan mengirim sinyal bahwa telah selesai menulis dan meninggalkan protokol.

Masalah di atas dapat dipandang sebagai sebuah SMA, dimana setiap agen dapat berfungsi sebagai pembaca ataupun penulis. Sebagai pembaca, sebuah agen  $k$  dapat melaksanakan aksi  $ImmRd(k)$ ,  $ReqRd(k)$ ,  $GrRd(k)$ , dan  $EndRd(k)$ ; sedangkan sebagai penulis sebuah agen dapat melaksanakan aksi  $ImmWr(k)$ ,  $ReqWr(k)$ ,  $GrWr(k)$  dan  $EndWr(k)$ . Sebagai kesimpulan, agen-agen tersebut bersifat homogen. Sehingga, kita dapat menulis spesifikasi dari sistem tersebut dengan menggunakan Formula 2.1, seperti ditunjukkan pada Gambar 3.1. Pada spesifikasi tersebut digunakan empat himpunan  $ar$ ,  $wr$ ,  $aw$  dan  $ww$  untuk menyatakan himpunan proses yang melakukan pembacaan, proses yang sedang menunggu untuk membaca, proses yang sedang menulis, dan yang sedang menunggu giliran untuk menulis.

$$\begin{aligned}
Init &\equiv ar = \{\} \wedge wr = \{\} \wedge aw = \{\} \wedge ww = \{\} \\
ImmRd(k) &\equiv \wedge k \notin ar \cup wr \cup aw \cup ww \wedge aw = \{\} \wedge ww = \{\} \wedge ar' = ar \cup \{k\} \\
&\quad \wedge UNCHANGED\langle wr, aw, ww \rangle \\
ReqRd(k) &\equiv \wedge k \notin ar \cup wr \cup aw \cup ww \wedge \neg(aw = \{\} \wedge ww = \{\}) \wedge wr' = wr \cup \{k\} \\
&\quad \wedge UNCHANGED\langle ar, aw, ww \rangle \\
GrRd(k) &\equiv \wedge k \in wr \wedge aw = \{\} \wedge ar' = ar \cup \{k\} \wedge wr' = wr \setminus \{k\} \\
&\quad \wedge UNCHANGED\langle aw, ww \rangle \\
EndRd(k) &\equiv \wedge k \in ar \wedge ar' = ar \setminus \{k\} \wedge UNCHANGED\langle wr, aw, ww \rangle \\
ImmWr(k) &\equiv \wedge k \notin ar \cup wr \cup aw \cup ww \wedge ar \cup wr \cup aw \cup ww = \{\} \wedge aw' = aw \cup \{k\} \\
&\quad \wedge UNCHANGED\langle ar, wr, ww \rangle \\
ReqWr(k) &\equiv \wedge k \notin ar \cup wr \cup aw \cup ww \wedge ww' = ww \cup \{k\} \\
&\quad \wedge ar \cup wr \cup aw \cup ww \neq \{\} \wedge UNCHANGED\langle ar, wr, aw \rangle \\
GrWr(k) &\equiv \wedge k \in ww \wedge aw = \{\} \wedge aw' = aw \cup \{k\} \\
&\quad \wedge ww' = ww \setminus \{k\} \wedge UNCHANGED\langle ar, wr \rangle \\
EndWr(k) &\equiv \wedge k \in aw \wedge aw' = aw \setminus \{k\} \wedge UNCHANGED\langle ar, wr, ww \rangle \\
Next(k) &\equiv \vee ImmRd(k) \vee ReqRd(k) \vee GrRd(k) \vee EndRd(k) \\
&\quad \vee ImmWr(k) \vee ReqWr(k) \vee GrWr(k) \vee EndWr(k) \\
v &\equiv \langle ar, wr, aw, ww \rangle \\
L(k) &\equiv WF_v(GrRd(k)) \wedge WF_v(EndRd(k)) \wedge SF_v(GrWr(k)) \wedge WF_v(EndWr(k)) \\
ParRW &\equiv \forall k \in M : Init \wedge \Box[Next(k)]_v \wedge L(k)
\end{aligned}$$

Gambar 3.1: Spesifikasi untuk *reader writer protocol*.



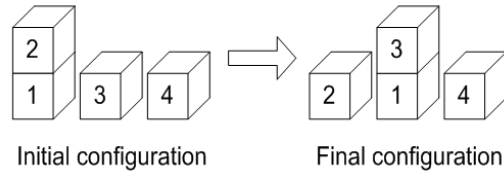


## Bab 4

# Studi Kasus 2: Block-world problem

Sebagai studi kasus yang kedua, diambil *block-world problem* dari [14]. Masalah ini dapat dijelaskan dengan singkat sebagai berikut: terdapat sekumpulan kotak di atas meja dan terdapat dua jenis robot yang bertugas untuk mengubah tumpukan kotak, dari susunan awal menjadi susunan baru yang berbeda. Setiap saat hanya boleh satu kotak yang diubah letaknya: apakah ditumpuk di atas kotak yang lain atau diambil dari tumpukan dan ditempatkan di atas meja. Kotak-kotak yang berada di bawah kotak yang lain tidak dapat diubah letaknya. Setiap robot mempunyai kemampuan yang berbeda: satu robot hanya dapat mengambil satu kotak dari tumpukan kotak, sementara robot yang satunya hanya dapat mengambil kotak dari atas meja dan menumpukkannya di atas kotak lainnya.

Gambar 4.1 menunjukkan contoh permasalahan ini. Terdapat empat kotak di atas meja. Setiap kotak diberi nomor. Gambar kiri menunjukkan susunan awal dan gambar kanan menunjukkan susunan akhir.



Gambar 4.1: Contoh dari *block-world problem*.

Terdapat sebuah masalah yang muncul jika kita akan menyatakan studi kasus ini sebagai sistem berparameter dengan menggunakan Formula 2.1. Untuk menggunakan Formula 2.1, disyaratkan bahwa sistem yang dinyatakan harus terdiri atas sejumlah agen yang sejenis. Pada kasus ini, agen, dalam hal ini robot, tidak bersifat homogen, karena mereka memiliki kemampuan yang berbeda.

Solusi yang diambil untuk masalah ini sangat sederhana. Untuk setiap robot diasosiasikan sebuah bilangan bulat yang menyatakan tipenya. Untuk memberikan kesan bahwa robot-robot tersebut mempunyai kemampuan yang berbeda, kita nyatakan kemampuan atau aksi yang dapat diambil oleh setiap robot secara terpisah. Untuk setiap aksi diberikan prekondisi. Prekondisi ini digunakan untuk menjamin bahwa hanya aksi tertentu yang dapat diambil oleh robot tertentu yang memang sesuai dengan kemampuannya. Syarat bahwa agen bersifat homogen masih dapat dipenuhi dengan menggunakan aksi *Next* yang sama untuk setiap agen.

Dengan mengasumsikan bahwa sistem terdiri atas empat kotak dan susunan awal dan akhir seperti ditunjukkan pada Gambar 4.1, spesifikasi dari *block-world problem* diberikan pada Gambar 4.2. Pada spesifikasi tersebut digunakan himpunan bilangan bulat, *Block*, untuk merepresentasikan kumpulan kotak dan sebuah larik *ag\_type* untuk mengidentifikasi tipe dari setiap agen.

Sebuah larik dari bilangan bulat tak negatif digunakan untuk merepresentasikan status dari susunan kotak. Sebagai contoh, jika elemen pertama larik adalah  $\langle 3, 2 \rangle$  maka berarti kotak nomor 1 berada di atas kotak nomor 3 dan di bawah kotak nomor

2. Untuk merepresentasikan meja atau tidak ada sesuatu di atas sebuah kotak digunakan bilangan 0. Susunan awal pada Gambar 4.1 direpresentasikan dengan  $\langle 0, 2 \rangle$ ,  $\langle 1, 0 \rangle$ ,  $\langle 0, 0 \rangle$ ,  $\langle 0, 0 \rangle$  dan susunan akhir dinyatakan sebagai  $\langle 0, 3 \rangle$ ,  $\langle 0, 0 \rangle$ ,  $\langle 1, 0 \rangle$ ,  $\langle 0, 0 \rangle$ .

Selain itu digunakan juga tiga larik lain yaitu *Istate*, *Cstate* dan *Fstate*. Setiap larik digunakan untuk mencatat susunan awal, akhir, dan susunan saat ini (yang sedang berlaku). *isDone* bernilai benar jika susunan saat ini sama dengan susunan akhir, yang berarti tujuan telah tercapai.

Nilai dari variabel sistem, yaitu *Block*, *ag\_type*, *Istate*, *Fstate*, tentu saja, tergantung pada instansi masalah yang sedang ditinjau. Modifikasi juga perlu dilakukan terhadap *Blocks* dan *ag\_type* sesuai dengan perubahan jumlah kotak dan tipe agen. Hal ini juga berlaku untuk kemampuan agen. Yang diperlukan adalah menambahkan prekondisi pada setiap aksi seperti yang telah dijelaskan sebelumnya.

$$\begin{aligned}
isDone &\equiv CState = FState \\
move(k) &\equiv \wedge \neg isDone \wedge ag\_type[k] = 1 \\
&\wedge \exists x, y \in Blocks : \\
&\quad \wedge x \neq y \\
&\quad \wedge CState[x] = \langle 0, 0 \rangle \wedge CState[y][2] = 0 \\
&\quad \wedge CState' = [CState \text{ EXCEPT } ![x][1] = y, ![y][2] = x] \\
free(k) &\equiv \wedge \neg isDone \wedge ag\_type[k] = 2 \\
&\wedge \exists x, y \in Blocks : \\
&\quad \wedge x \neq y \\
&\quad \wedge CState[x][2] = 0 \wedge CState[y][2] = x \\
&\quad \wedge CState' = [CState \text{ EXCEPT } ![x] = \langle 0, 0 \rangle, ![y][2] = 0] \\
Init(k) &\equiv CState = IState \wedge CState \neq FState \wedge ag\_type[k] \in \{1, 2\} \\
Next(k) &\equiv Move(k) \vee Free(k) \\
L(k) &\equiv WF_v(Move(k)) \wedge WF_v(Free(k)) \\
v &\equiv \langle CState \rangle \\
BWorld &\equiv \forall k \in M : Init(k) \wedge \Box[Next(k)]_v \wedge L(k)
\end{aligned}$$

Gambar 4.2: Spesifikasi untuk block-world problem.

## Bab 5

# Penutup

Salah satu syarat sebuah SMA dapat dinyatakan sebagai suatu sistem berparameter dengan menggunakan spesifikasi standar TLA adalah sistem tersebut harus bersifat homogen. Tujuan dari penelitian ini adalah melihat sejauh mana sebuah SMA dapat dinyatakan sebagai suatu sistem berparameter.

Pada penelitian ini telah berhasil dibuat spesifikasi dua studi kasus, yaitu *reader write protocol* dan *block-world problem*. Dua studi kasus ini berbeda dalam hal homogenitas dari agennya. Kasus yang pertama adalah contoh SMA homogen, sementara kasus kedua adalah contoh SMA non-homogen. Agar memenuhi syarat homogen, untuk kasus yang kedua, ditambahkan prekondisi pada aksi dalam spesifikasi untuk menjamin bahwa agen yang tepat yang bisa mengaktifkan aksi tertentu sesuai dengan deskripsi masalah.

Terdapat banyak penelitian lain tentang spesifikasi dari SMA. Biasanya, fokus dari penelitian tersebut adalah spesifikasi dari perilaku agen dan koordinasi antar agen. Salah satu penelitian tentang spesifikasi SMA menggunakan Petri net, dengan mengambil studi kasus sistem transportasi telah dilakukan oleh Abouissa dkk. di [1]. Sebuah formalismus lain, disebut Utility State Machines untuk pembuatan

spesifikasi SMA pada aplikasi *e-commerce* telah diusulkan oleh Merayom dkk. di [9]. Brazier dkk. (Brazier et al, 1995) menggunakan *framework* DESIRE, untuk membuat spesifikasi aplikasi SMA dunia nyata pada level konseptual. Fischer dan Wooldridge menggunakan logika temporal untuk mendeskripsikan spesifikasi formal dari SMA [4]. Penelitian ini mirip dengan yang dilakukan oleh Taibi [14], yaitu menggunakan TLA sebagai bahasa spesifikasi. Perbedaannya adalah bahwa Taibi dkk. tidak meninjau SMA sebagai suatu sistem berparameter.

Sebagai rencana ke depan, direncanakan untuk mengembangkan penelitian ini dengan:

1. mengurangi pembatasan permasalahan untuk studi kasus yang telah dilakukan, dengan meninjau sifat-sifat SMA yang pada penelitian ini masih diabaikan.
2. meninjau studi kasus yang lebih kompleks untuk memperkaya dan menguatkan kesimpulan yang telah diperoleh pada penelitian ini.

Sebagai catatan, hasil dari penelitian ini telah dipublikasikan pada sebuah konferensi internasional yaitu SEAMS-GMU 2011.

# Referensi

- [1] ABOUAISSA, H. ET AL, Formal specification of multi-agent systems: approach based on meta-models and high-level Petri nets - case study of a transportation system, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, 2002.
- [2] AYED, B. AND SIALA, F., Event-B based Verification of Interaction Properties In Multi-Agent Systems, *Journal of Software*, vol. 4, no. 4, 2009.
- [3] BRAZIER, F.M.T, ET AL., Formal Specification of Multi-Agent Systems: a Real World Case, *Proceedings of the First International Conference on Multi-Agent Systems*, 1995.
- [4] FISHER, M. AND WOOLDRIDGE, M., the Formal Specification and Verification of Multi-Agent Systems, *Int. J. Cooperative Inf. Syst.*, 37-66, 1997.
- [5] GAUD, N., A Verification by Abstraction Framework for organizational Multi-Agent Systems. *Jung, Michel, Ricci & Petta (eds.): AT2AI-6 Working Notes, From Agent Theory to Agent Implementation*, 6th, 2008.
- [6] GIESE, H. AND KLEIN, F., Systematic verification of multi-agent systems based on rigorous executable specifications. *Journal International Journal of Agent-Oriented Software Engineering*, Volume 1 Issue 1, 2007.

- [7] LAMPORT, L., The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, Vol. 16, No. 3, pp.872-923, 1994.
- [8] MASSACCI, F., Tableau Methods for Formal Verification of Multi-Agent Distributed Systems. *Journal of Logic and Computation*, **8(3)**, 1998.
- [9] MERAYON, M.G. ET.AL., Formal specification of multi-agent systems by using EUSMs. *Proceedings of International Conference on Fundamentals of Software Engineering*, 2007.
- [10] NUGRAHENI, C.E., Mutual Exclusion Verification of Parameterized Reader-Writer Algorithm: a case study. *Proceeding of Seminar Nasional Aplikasi Teknologi Informasi*, 2006.
- [11] NUGRAHENI, C.E., Diagram-based verification of parameterized systems. *JCMCC* **65**, pp. 91-102, 2008.
- [12] NUGRAHENI, C.E., Formal Verification of Parameterized Multi-Agent Systems using Predicate Diagrams\*. *Proceeding of COMPUTATION TOOLS 2011*, to appear.
- [13] SYCARA, K.P., Multiagent Systems. *AI Magazine*, vol. 19, no. 2, 1998.
- [14] TAIBI, T., Formal specification and validation of multi-agent behaviour using TLA+ and TLC model checker. *Int. J. Artificial Intelligence and Soft Computing*, vol. 1, no. 1, 2008.
- [15] SZIRBIK, N. AND WAGNER, G. Steps Towards Formal Verification of Agent-based E-Business Applications. Dept. of Information & Technology, Eindhoven University of Technology Postbus 513, 5600 MB Eindhoven. Research preliminary report.